

ON ASYMPTOTIC GATE COMPLEXITY AND DEPTH OF REVERSIBLE CIRCUITS WITH ADDITIONAL MEMORY

DMITRY V. ZAKABLUKOV

March 22, 2016

Abstract. The reversible logic can be used in various research areas, e. g. quantum computation, cryptography and signal processing. In the paper we study reversible logic circuits with additional inputs, which consist of NOT, CNOT and C²NOT gates. We consider a set $F(n, q)$ of all transformations $\mathbb{B}^n \rightarrow \mathbb{B}^n$ that can be realized by reversible circuits with $(n + q)$ inputs. An analogue of Lupanov's method for the synthesis of reversible logic circuits with additional inputs is described. We prove upper asymptotic bounds for the Shannon gate complexity function $L(n, q)$ and the depth function $D(n, q)$ in case of $q > 0$: $L(n, q_0) \lesssim 2^n$ if $q_0 \sim n2^{n-o(n)}$ and $D(n, q_1) \lesssim 3n$ if $q_1 \sim 2^n$.

Keywords. Reversible logic, gate complexity, circuit depth, asymptotic bounds.

Subject classification. 03D15 Complexity of computation.

1. Introduction

The reversible logic is essential in the quantum computing. It also has a great potential in designing various computing devices with low power consumption. Landauer (1961) proved that the irreversibility of computations leads to the energy dissipation regardless of the underlying technology. Bennett (1973) showed that the absence of heat generation can be achieved only when a circuit is completely built from reversible gates. The main problem is that we should find a compromise between the gate complexity,

the depth (working time) of a reversible circuit and the amount of used memory (additional inputs) when solving the problem of reversible logic synthesis. Unfortunately, strict asymptotic bounds for the gate complexity and the depth of reversible circuits haven't been found so far, especially in the case of using additional inputs.

The circuit complexity theory goes back to the work of Shannon (1949). He was the first who suggested to consider the complexity of the minimal switching circuit, which realizes a Boolean function, as a complexity measure of this function. For today, the asymptotic gate complexity $L(n) \sim 2^n / n$ of a Boolean function of n variables in the basis of classical gates “NOT, OR, AND” is well-known.

The problem of computations with the limited memory was considered by Karpova (1987). She proved that the asymptotic gate complexity of a circuit, which consists of the gates corresponding to all Boolean functions of p variables and which uses at least three memory registers, depends on the value of p , but doesn't depend on the number of used memory registers. Also she proved that any Boolean function can be realized in such a circuit using only two memory registers.

Lupanov (1970) considered circuits of functional elements with delays. He proved that in a regular basis of functional elements any Boolean function can be realized in a circuit with asymptotically the best gate complexity and with the delay $T(n) \sim \tau n$, where τ is the constant depending on the basis. Though the depth and the delay of a circuit can be defined differently (see Khrapchenko 1995), in the model of reversible circuit described below we can consider the value of $T(n)$ as the circuit depth. However a dependency of $T(n)$ on the number of used memory registers was not considered for the “classical” circuits.

A gate is called reversible if it implements a bijective transformation. There are several known reversible gates for today. Among them are the NOT gate; the controlled NOT (CNOT) gate, introduced by Feynman (1985); the Toffoli gate (C²NOT) introduced by Toffoli (1980); the Fredkin gate, etc.

A set $F(n, q)$ of all transformations $\mathbb{B}^n \rightarrow \mathbb{B}^n$ that can be implemented by reversible circuits with $(n + q)$ inputs was considered in Zakablukov (2015). Also the Shannon gate complexity function

$L(n, q)$ and the depth function $D(n, q)$ as functions of n and the number of additional inputs q (additional memory) were defined and upper bounded in the case, when additional inputs are not allowed in a reversible circuit.

The subject of this paper is reversible logic circuits, which consist of NOT, CNOT and C²NOT gates and which can use an unlimited amount of additional inputs (unlike the reversible circuits we have studied earlier, see Zakablukov 2015).

We will describe an analogue of Lupanov's method for synthesizing a reversible circuit with additional inputs, which has the minimal gate complexity or the minimal depth. Using this synthesis approach, we will prove the following upper asymptotic bounds for the functions $L(n, q)$ and $D(n, q)$:

$$\begin{aligned} L(n, q_0) &\lesssim 2^n, \text{ if } q_0 \sim n2^{n-o(n)}, \\ D(n, q_1) &\lesssim 3n, \text{ if } q_1 \sim 2^n. \end{aligned}$$

Also, some upper bounds for the quantum weight function will be proved.

Using the lower and upper bounds for the functions $L(n, q)$ and $D(n, q)$, we state that the usage of additional memory in a reversible circuit, which consists of NOT, CNOT and C²NOT gates, almost always allows to reduce its gate complexity and the depth.

2. Background

The controlled NOT gate (CNOT) was introduced by Feynman (1985). The Toffoli gate was introduced by Toffoli (1980). The generalized Toffoli gate with multiple control inputs is usually denoted as C^kNOT or TOF_{*k*+1}, where k stands for the number of control inputs. The synthesis of reversible circuits consisting of these gates was discussed in several works, see Khlopotine *et al.* (2002); Maslov *et al.* (2007); Miller & Dueck (2003); Miller *et al.* (2003); Saeedi *et al.* (2007, 2010); Zakablukov (2014).

We use the following notation for a generalized Toffoli gate.

DEFINITION 2.1. *A generalized Toffoli gate with k control inputs $TOF_{k+1}^n(I; t) = TOF_{k+1}^n(i_1, \dots, i_k, t)$ is a reversible gate with n*

inputs, which defines a transformation $\mathbb{B}^n \rightarrow \mathbb{B}^n$ as follows:

$$f_{TOF_{k+1}^n(I;t)}(\langle x_1, \dots, x_n \rangle) = \langle x_1, \dots, x_t \oplus x_{i_1} \wedge \dots \wedge x_{i_k}, \dots, x_n \rangle ,$$

where $I = \{i_1, \dots, i_k\}$ is a set of indices of control input lines and t is an index of a controlled output line, $t \notin I$.

From the definition one can note that a gate $TOF(a)$ is a NOT gate, $TOF(a,b)$ is a CNOT gate and $TOF(a,b,c)$ is a C^2 NOT gate.

We denote a set of all TOF_{k+1}^n gates, $k < 3$, as Ω_n^2 (i.e. all NOT, CNOT and C^2 NOT gates). An upper and/or a lower indices in TOF_{k+1}^n will be omitted, if their value is clear from the context.

Fan-in, fan-out and a random connection of inputs and outputs of gates in a reversible circuit are forbidden. We assume that all gates in a reversible circuit have exactly n numbered inputs and outputs and that the i -th output of a gate is connected only to the i -th input of the following gate. Thus in our model of a reversible circuit a graph associated with a circuit presents itself a single chain. We will refer to such a connection of reversible gates as *composition*.

A symbol r_i from a set $R = \{r_1, \dots, r_n\}$ can be assigned to the i -th input and output of a gate. All these symbols can be treated as names of memory registers (indices of memory cells), which store the current computation result of a circuit.

If we consider all the gates from Ω_n^2 regardless of an underlying technology, we can assume that they all have the same technological cost. However, in a quantum technology, for example, a technological cost of NOT and CNOT gates is much less than a technological cost of a Toffoli gate (see Barenco *et al.* 1995). Hence, we will assume that a gate e from Ω_n^2 has the weight $W(e)$ depending on the underlying technology. More precisely, we will assume that all NOT and CNOT gates from Ω_n^2 have the same weight $W^{(C)}$ and all C^2 NOT gates from Ω_n^2 have the weight $W^{(T)}$.

Let a reversible circuit \mathfrak{S} with n inputs be a composition of l gates from Ω_n^2 : $\mathfrak{S} = *_{j=1}^l TOF(I_j; t_j)$. In the paper we study the following circuit's properties:

1. The gate complexity $L(\mathfrak{S})$, equal to the number of gates l .

2. The quantum weight $W(\mathfrak{S})$, equal to the sum of weights of all its gates.
3. The depth $D(\mathfrak{S})$, equal to the number of gates in the path from inputs to outputs that cannot be executed simultaneously.

Note that the quantum weight $W(\mathfrak{S})$ of a reversible circuit \mathfrak{S} is not equal to its technological cost, because they may significantly differ. But we can state that in most cases a greater value of the function $W(\mathfrak{S})$ means a greater technological cost of a reversible circuit \mathfrak{S} .

A formal definition of the reversible circuit depth for our circuit model can be found in Zakablukov (2015). Here we just want to remind that a reversible circuit \mathfrak{S} has the depth $D(\mathfrak{S}) = 1$, if for every two of its gates $TOF(I_1; j_1)$ and $TOF(I_2; j_2)$ the following equation holds:

$$(\{t_1\} \cup I_1) \cap (\{t_2\} \cup I_2) = \emptyset.$$

Also, the depth $D(\mathfrak{S})$ of a reversible circuit \mathfrak{S} equals to the minimal number d of disjoint sub-circuits with the depth of each equal to one in the following equation:

$$\mathfrak{S} = \bigsqcup_{i=1}^d \mathfrak{S}'_i, \quad \mathfrak{S}'_i \subseteq \mathfrak{S}, \quad D(\mathfrak{S}'_i) = 1.$$

For example, a reversible circuit $\mathfrak{S} = TOF(1; 2) * TOF(3, 1) * TOF(2) * TOF(4) * TOF(1, 4, 2) * TOF(3)$ (see Figure 2.1) has the gate complexity $L(\mathfrak{S}) = 6$ and the depth $D(\mathfrak{S}) = 3$, because we can divide the circuit into three disjoint sub-circuits with the depth of each equal to one in the following manner: $\mathfrak{S} = (TOF(1; 2)) * (TOF(3, 1) * TOF(2) * TOF(4)) * (TOF(1, 4, 2) * TOF(3))$.

Note that the reversible circuit is equivalent to another one with the depth equal to three: $\mathfrak{S}_1 = (TOF(1; 2) * TOF(4)) * (TOF(3, 1) * TOF(2)) * (TOF(1, 4, 2) * TOF(3))$. Therefore from here on we will consider that such circuits are different in terms of our reversible circuit's model, but equivalent in terms of the equality of Boolean transformations defined by them.

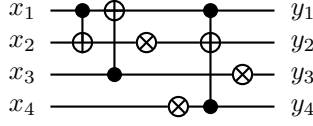


Figure 2.1: A reversible circuit $\mathfrak{S} = TOF(1; 2) * TOF(3, 1) * TOF(2) * TOF(4) * TOF(1, 4, 2) * TOF(3)$ with the gate complexity $L(\mathfrak{S}) = 6$ and the depth $D(\mathfrak{S}) = 3$.

3. Asymptotic bounds for reversible circuits without additional inputs

It is well-known that a reversible circuit \mathfrak{S} with $n \geq 4$ inputs defines an even permutation on the set \mathbb{B}^n , see Shende *et al.* (2003). But it can also implement a transformation $\mathbb{B}^m \rightarrow \mathbb{B}^k$, where $m, k \leq n$, with or without an additional memory. A circuit with $(n+q)$ inputs implements a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^m$, if there is such a permutation $\pi \in S(\mathbb{Z}_{n+q})$ for circuit outputs that every input of the form $\langle x_1, \dots, x_n, 0, \dots, 0 \rangle$ is transformed by the circuit into an output $\langle y_1, \dots, y_m, *, \dots, * \rangle$ after applying the permutation π , where $f(\langle x_1, \dots, x_n \rangle) = \langle y_1, \dots, y_m \rangle$ (see Figure 3.1).

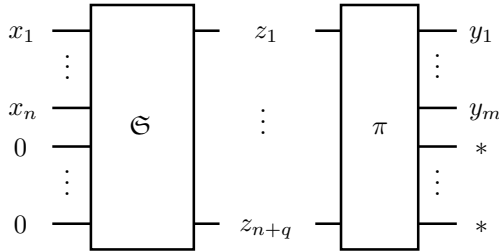


Figure 3.1: A reversible circuit \mathfrak{S} implementing a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^m$ with q additional inputs. For every $\mathbf{x} \in \mathbb{B}^n$ the equation $f(\langle x_1, \dots, x_n \rangle) = \langle y_1, \dots, y_m \rangle$ holds.

We remind that in our terminology expressions “implements a transformation” and “defines a transformation” have different meanings. If a circuit \mathfrak{S} implements a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^n$ and has exactly n inputs, we will say that this circuit implements f *without additional inputs*.

We marked all “don’t care” outputs of a reversible circuit by the symbol $*$ on Figure 3.1. In most cases these outputs will not be cleared out in the end, i.e. they will contain a *computational garbage*. Unfortunately, this garbage can be removed only if a transformation f implemented by a reversible circuit \mathfrak{S} is bijective. In this case we can clear out all garbage outputs, except ones corresponding to the inputs of f , with the help of a part of the existing circuit (let’s denote it as \mathfrak{S}_*). Then we can append a reversible circuit \mathfrak{S}^{-1} implementing the transformation f^{-1} with generating of computational garbage and with clearing out the outputs corresponding to the inputs of f . And finally, we can remove this generated garbage with the help of a part of the circuit \mathfrak{S}^{-1} (let’s denote it as \mathfrak{S}_*^{-1}). Thus a resulting circuit $\mathfrak{S}_{\text{res}}$ will have the gate complexity $L(\mathfrak{S}_{\text{res}}) \leq 4 \cdot \max(L(\mathfrak{S}), L(\mathfrak{S}^{-1}))$ and the depth $D(\mathfrak{S}_{\text{res}}) \leq 4 \cdot \max(D(\mathfrak{S}), D(\mathfrak{S}^{-1}))$ (see Figure 3.2).

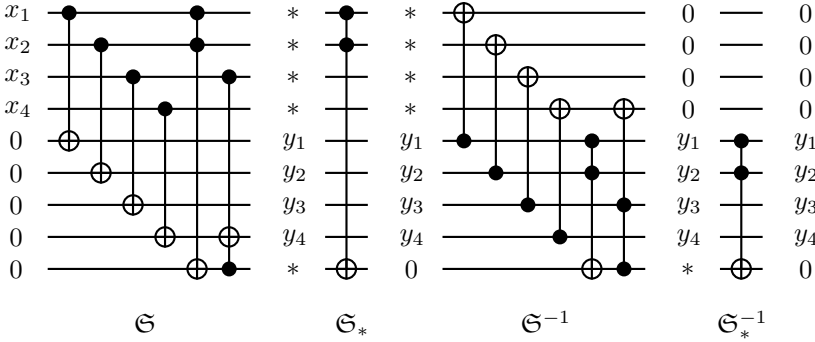


Figure 3.2: An example of a reversible circuit $\mathfrak{S}_{\text{res}} = \mathfrak{S} * (\mathfrak{S}_*) * \mathfrak{S}^{-1} * (\mathfrak{S}_*^{-1})$ with garbage removing.

Therefore, all asymptotic bounds for the gate complexity and the depth will be given later for a reversible circuit with a computational garbage on the outputs. To obtain similar bounds for a reversible circuit without a computational garbage on the outputs, one should multiply them by four.

Let $P_2(n, n)$ be the set of all transformations $\mathbb{B}^n \rightarrow \mathbb{B}^n$ and $F(n, q) \subseteq P_2(n, n)$ be the set of all transformations, which can be implemented by reversible circuits with $(n + q)$ inputs. It is not difficult to show that $F(n, 0)$, $n > 3$, is equal to the set of

transformations that are defined by all the permutations from the alternating group $A(\mathbb{B}^n)$ and $F(n, q) = P_2(n, n)$ if $q \geq n$.

We denote the minimum gate complexity, the minimum depth and the minimum quantum weight of a reversible circuit among all reversible circuits implementing a transformation $f \in F(n, q)$ with q additional inputs as $L(f, q)$, $D(f, q)$ and $W(f, q)$ respectively. The Shannon gate complexity function $L(n, q)$, the depth function $D(n, q)$ and the quantum weight function $W(n, q)$ are defined as follows:

$$\begin{aligned} L(n, q) &= \max_{f \in F(n, q)} L(f, q) , \\ D(n, q) &= \max_{f \in F(n, q)} D(f, q) , \\ W(n, q) &= \max_{f \in F(n, q)} W(f, q) . \end{aligned}$$

For the purpose of estimating the function $W(n, q)$, we will count the number of NOT/CNOT and C^2 NOT gates in a reversible circuit separately. If we denote the number of NOT and CNOT gates in a reversible circuit \mathfrak{S} as $L^{(C)}(\mathfrak{S})$ and the number of C^2 NOT gates as $L^{(T)}(\mathfrak{S})$, then the following equation holds:

$$(3.1) \quad W(\mathfrak{S}) = W^{(C)} \cdot L^{(C)}(\mathfrak{S}) + W^{(T)} \cdot L^{(T)}(\mathfrak{S}) .$$

We proved (see Zakablukov 2015) that there is such $n_0 \in \mathbb{N}$ that for $n > n_0$ the following equations hold:

$$(3.2) \quad L(n, q) \geq \frac{2^n(n-2)}{3 \log_2(n+q)} - \frac{n}{3} ,$$

$$(3.3) \quad D(n, q) \geq \frac{2^n(n-2)}{3(n+q) \log_2(n+q)} - \frac{n}{3(n+q)} ,$$

$$(3.4) \quad W(n, q) \geq \min(W^{(C)}, W^{(T)}) \cdot \left(\frac{2^n(n-2)}{3 \log_2(n+q)} - \frac{n}{3} \right) .$$

Also, the following upper bounds for a reversible circuit without

additional inputs were proved (see Zakablukov 2015):

$$(3.5) \quad L(n, 0) \leq \frac{3n2^{n+4}}{\log_2 n - \log_2 \log_2 n - \log_2 \phi(n)} (1 + \epsilon_L(n)) ,$$

$$(3.6) \quad D(n, 0) \leq \frac{n2^{n+5}}{\log_2 n - \log_2 \log_2 n - \log_2 \phi(n)} (1 + \epsilon_D(n)) ,$$

$$(3.7) \quad W(n, 0) \leq \frac{n2^{n+4} (W^{(C)}(1 + \epsilon_C(n)) + 2W^{(T)}(1 + \epsilon_T(n)))}{\log_2 n - \log_2 \log_2 n - \log_2 \phi(n)} ,$$

where $\phi(n) < n / \log_2 n$ is an arbitrarily slowly growing function and

$$\begin{aligned} \epsilon_L(n) &= \frac{1}{6\phi(n)} + \left(\frac{8}{3} - o(1) \right) \frac{\log_2 n \cdot \log_2 \log_2 n}{n} , \\ \epsilon_D(n) &= \frac{1}{4\phi(n)} + (4 - o(1)) \frac{\log_2 n \cdot \log_2 \log_2 n}{n} , \\ \epsilon_C(n) &= \frac{1}{2\phi(n)} - \left(\frac{1}{2} - o(1) \right) \cdot \frac{\log_2 \log_2 n}{n} , \\ \epsilon_T(n) &= (4 - o(1)) \frac{\log_2 n \cdot \log_2 \log_2 n}{n} . \end{aligned}$$

Unfortunately, there are no known upper asymptotic bounds for the functions $L(n, q)$, $D(n, q)$ and $W(n, q)$ in the case when a reversible circuit can use an unlimited amount of additional inputs for today. Nevertheless, it has been already showed that in some cases the usage of additional memory in a reversible circuit consisting of gates from Ω_n^2 allows to reduce its gate complexity and the depth, see Abdessaied *et al.* (2013); Barenco *et al.* (1995); Miller *et al.* (2010).

4. Reducing the gate complexity with the help of additional inputs

Lupanov described asymptotically the best synthesis algorithm of a Boolean function in the basis $\{\neg, \wedge, \vee\}$. He proved that any Boolean function of n variables can be implemented in a circuit with the gate complexity $L \sim 2^n / n$ and with the total delay no more than $O(n)$, see Lupanov (1970). We will modify Lupanov's

method in order to synthesize a reversible circuit, which consists of gates from Ω_{n+q}^2 and implements a transformation $f \in F(n, q)$ with q additional inputs.

The basis $\{ \neg, \oplus, \wedge \}$ is functionally complete, therefore it can be used to implement any transformation $f \in F(n, q)$. Let's express every element of this basis via a composition of NOT, CNOT and C²NOT gates (see Figure 4.1). As we can see, this requires no more than two gates and one additional input for every element of the basis.

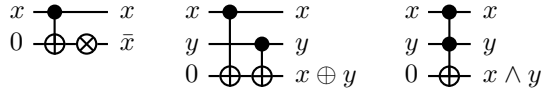


Figure 4.1: Implementing elements of the basis $\{ \neg, \oplus, \wedge \}$ with compositions of NOT, CNOT and C²NOT gates.

First, we prove the following lemma about the gate complexity of a reversible circuit implementing all conjunctions of n variables of the form $x_1^{a_1} \wedge \dots \wedge x_n^{a_n}$, $a_i \in \mathbb{B}$.

LEMMA 4.1. *All conjunctions of n variables of the form $x_1^{a_1} \wedge \dots \wedge x_n^{a_n}$, $a_i \in \mathbb{B}$, can be implemented in a reversible circuit \mathfrak{S}_n , which consists of gates from Ω_{n+q}^2 , with the gate complexity $L(\mathfrak{S}_n) \sim 2^n$ and with $q(\mathfrak{S}_n) \sim 2^n$ additional inputs.*

PROOF. First step is obtaining inversions of all input variables: \bar{x}_i , $1 \leq i \leq n$. This can be done using $L_1 = 2n$ NOT and CNOT gates and $q_1 = n$ additional inputs.

We construct our reversible circuit \mathfrak{S}_n this way: using circuits $\mathfrak{S}_{\lceil n/2 \rceil}$ and $\mathfrak{S}_{\lfloor n/2 \rfloor}$, we implement all conjunctions of the first $\lceil n/2 \rceil$ and the last $\lfloor n/2 \rfloor$ variables (see Figure 4.2). After this we implement conjunctions of outputs of the circuit $\mathfrak{S}_{\lceil n/2 \rceil}$ with outputs of the circuit $\mathfrak{S}_{\lfloor n/2 \rfloor}$. This can be done using $L_2 = 2^n$ C²NOT gates and $q_2 = 2^n$ additional inputs.

Hence, the following equation holds:

$$L(\mathfrak{S}_n) \sim q(\mathfrak{S}_n) \sim 2^n + 2L(\mathfrak{S}_{n/2}) \sim 2^n .$$

□

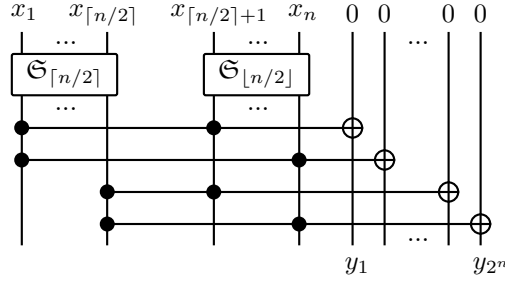


Figure 4.2: The structure of a reversible circuit \mathfrak{S}_n implementing conjunctions of n variables with the minimal gate complexity.

Now we can prove the first theorem of the paper.

THEOREM 4.2.

$$L(n, q_0) \lesssim 2^n, \text{ if } q_0 \sim n2^{n-\lceil n/\phi(n) \rceil},$$

where $\phi(n) \leq n/(\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions.

PROOF. We will describe a new synthesis algorithm **A1**, which is similar to the Lupanov's method and whose main goal is the reduction of the gate complexity with the help of additional inputs.

Let's consider a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^n$. It can be represented as follows:

$$(4.3) \quad f(\mathbf{x}) = \bigoplus_{a_{k+1}, \dots, a_n \in \mathbb{B}} x_{k+1}^{a_{k+1}} \wedge \dots \wedge x_n^{a_n} \wedge f(\langle x_1, \dots, x_k, a_{k+1}, \dots, a_n \rangle).$$

Each of 2^{n-k} Boolean transformations

$$f(\langle x_1, \dots, x_k, a_{k+1}, \dots, a_n \rangle) = f_i(\langle x_1, \dots, x_k \rangle),$$

where $\sum_{j=1}^{n-k} a_{k+j}2^{j-1} = i$, is a Boolean transformation $\mathbb{B}^k \rightarrow \mathbb{B}^n$ and can be represented as the system of n coordinate functions $f_{i,j}(\mathbf{x})$, $\mathbf{x} \in \mathbb{B}^k$, $1 \leq j \leq n$.

The value of every coordinate function $f_{i,j}(\mathbf{x})$ can be calculated with the help of an analogue of a disjunctive normal form:

$$(4.4) \quad f_{i,j}(\mathbf{x}) = \bigoplus_{\substack{\sigma \in \mathbb{B}^k \\ f_{i,j}(\sigma)=1}} x_1^{\sigma_1} \wedge \cdots \wedge x_k^{\sigma_k}.$$

All 2^k conjunctions of the form $x_1^{\sigma_1} \wedge \cdots \wedge x_k^{\sigma_k}$ can be divided into the groups with no more than s conjunctions in each. The number of such groups is $p = \lceil 2^k / s \rceil$. Using conjunctions of a single group, we can construct no more than 2^s Boolean functions by the formula ((4.4)).

Let G_i be the set of all Boolean functions that can be constructed with the help of conjunctions of an i -th group, $1 \leq i \leq p$. Then $|G_i| \leq 2^s$. Therefore, we can rewrite equation ((4.4)) as follows:

$$(4.5) \quad f_{i,j}(\mathbf{x}) = \bigoplus_{\substack{t=1 \dots p \\ g_{j_t} \in G_t \\ 1 \leq j_t \leq |G_t|}} g_{j_t}(\mathbf{x}).$$

Note that all Boolean functions of a group G_i can be implemented, using a similar technique as in the Lemma 4.1. From the Figure 4.2 we can see that all C²NOT gates will be simply replaced by compositions of two CNOT gates. Thus, $L \sim 2^{s+1}$ CNOT gates and $q \sim 2^s$ additional inputs are required for this part in total.

The synthesis algorithm **A1** constructs a reversible circuit \mathfrak{S} implementing the transformation f ((4.3)) from the following sub-circuits (see Figure 4.3):

1. Sub-circuit \mathfrak{S}_1 implementing all conjunctions of the first k variables x_i by the Lemma 4.1 with the gate complexity $L_1 \sim 2^k$ and with $q_1 \sim 2^k$ additional inputs. The sub-circuit almost completely consists of C²NOT gates (the number of other gates is negligible).
2. Sub-circuit \mathfrak{S}_2 implementing all Boolean functions $g \in G_i$ for all $i \in \mathbb{Z}_p$ by the formula ((4.4)) with the gate complexity

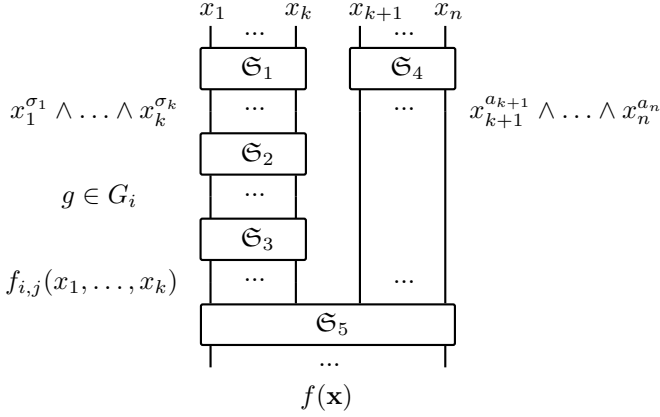


Figure 4.3: The structure of a reversible circuit \mathfrak{S} produced by the synthesis algorithm **A1**.

$L_2 \sim p2^{s+1}$ and with $q_2 \sim p2^s$ additional inputs (see the note above about the implementation of all Boolean functions of a group G_i). The sub-circuit consists only of CNOT gates.

3. Sub-circuit \mathfrak{S}_3 implementing all $n2^{n-k}$ coordinate functions $f_{i,j}(\mathbf{x})$, $i \in \mathbb{Z}_{2^{n-k}}$, $j \in \mathbb{Z}_n$, by the formula ((4.5)) with the gate complexity $L_3 \leq pn2^{n-k}$ and with $q_3 = n2^{n-k}$ additional inputs. The sub-circuit consists only of CNOT gates.
4. Sub-circuit \mathfrak{S}_4 implementing all conjunctions of the last $(n - k)$ variables x_i by the Lemma 4.1 with the gate complexity $L_4 \sim 2^{n-k}$ and with $q_4 \sim 2^{n-k}$ additional inputs. The sub-circuit almost completely consists of C^2 NOT gates (the number of other gates is negligible).
5. Sub-circuit \mathfrak{S}_5 implementing the transformation f by the formula ((4.3)) with the gate complexity $L_5 \leq n2^{n-k}$ and with $q_5 = n$ additional inputs. The sub-circuit consists only of C^2 NOT gates.

We are seeking the values of parameters k and s that satisfy

the following conditions:

$$\left\{ \begin{array}{l} s = n - 2k , \\ k = \lceil n / \phi(n) \rceil , \quad \text{where } \phi(n) \text{ is a growing function,} \\ 1 \leq s < n , \\ 1 \leq k < n / 2 , \\ 2^k / s \geq \psi(n) , \quad \text{where } \psi(n) \text{ is a growing function.} \end{array} \right.$$

In this case $p = \lceil 2^k / s \rceil \sim 2^k / s$ and $2^{\lceil n / \phi(n) \rceil} \geq s\psi(n)$. This implies that for any growing function $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ the values of parameters k and s will satisfy the conditions above.

Summing up gate complexities and the number of additional inputs of sub-circuits \mathfrak{S}_1 – \mathfrak{S}_5 , we obtain the following bounds:

$$L(\mathfrak{S}) \sim 2^k + p2^{s+1} + pn2^{n-k} + 2^{n-k} + n2^{n-k} ,$$

$$L(\mathfrak{S}) \sim 2^k + \frac{2^{n-k+1}}{s} + \frac{n2^n}{s} ,$$

$$q(\mathfrak{S}) \sim 2^k + p2^s + n2^{n-k} + 2^{n-k} + n \sim 2^k + \frac{2^{n-k}}{s} + n2^{n-k} .$$

Hence, if $k = \lceil n / \phi(n) \rceil$ and $s = n - 2k$, where $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are growing functions, the following equations hold:

$$L(\mathfrak{S}) \sim 2^{\lceil n / \phi(n) \rceil} + \frac{2^{n+1}}{n(1 - o(1))2^{\lceil n / \phi(n) \rceil}} + \frac{n2^n}{n(1 - o(1))} \sim 2^n ,$$

$$q(\mathfrak{S}) \sim 2^{\lceil n / \phi(n) \rceil} + \frac{2^n}{n(1 - o(1))2^{\lceil n / \phi(n) \rceil}} + \frac{n2^n}{2^{\lceil n / \phi(n) \rceil}} \sim \frac{n2^n}{2^{\lceil n / \phi(n) \rceil}} .$$

Since the synthesis algorithm **A1** can produce a reversible circuit \mathfrak{S} for any Boolean transformation $f \in F(n, q)$, we can state that $L(n, q_0) \leq L(\mathfrak{S}) \sim 2^n$, if $q_0 \sim n2^{n - \lceil n / \phi(n) \rceil}$, where $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions. \square

THEOREM 4.6.

$$L(n, q_0) \asymp 2^n, \quad \text{if } q_0 \sim n2^{n - \lceil n / \phi(n) \rceil} ,$$

where $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions.

PROOF. Follows from the lower bound ((3.2)) for the function $L(n, q)$ and from the Theorem 4.2. \square

Now we can upper bound the quantum weight of a reversible circuit with additional inputs.

THEOREM 4.7.

$W(n, q_0) \lesssim W^{(C)} \cdot 2^n + W^{(T)} \cdot n2^{n-\lceil n/\phi(n) \rceil}$, if $q_0 \sim n2^{n-\lceil n/\phi(n) \rceil}$, where $\phi(n) \leq n/(\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions.

PROOF. To prove the bound of the theorem, we should count the number of NOT, CNOT and C²NOT gates in a reversible circuit produced by the synthesis algorithm **A1**.

From the description of the algorithm we can see that

$$\begin{aligned} L_1^{(C)} &= O(k), & L_1^{(T)} &\sim 2^k, \\ L_2^{(C)} &\sim p2^{s+1}, & L_2^{(T)} &= 0, \\ L_3^{(C)} &\leq pn2^{n-k}, & L_3^{(T)} &= 0, \\ L_4^{(C)} &= O(n-k), & L_4^{(T)} &\sim 2^{n-k}, \\ L_5^{(C)} &= 0, & L_5^{(T)} &\leq n2^{n-k}. \end{aligned}$$

Providing $k = \lceil n/\phi(n) \rceil$ and $s = n - 2k$, where $\phi(n) \leq n/(\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are growing functions, we obtain the following upper bounds:

$$\begin{aligned} L^{(C)}(n, q_0) &\lesssim O(k) + p2^{s+1} + pn2^{n-k} + O(n-k), \\ L^{(C)}(n, q_0) &\lesssim \frac{2^{k+s+1}}{s} + \frac{n2^n}{s}, \\ L^{(C)}(n, q_0) &\lesssim \frac{2^{n+1}}{(n - o(n))2^{\lceil n/\phi(n) \rceil}} + \frac{n2^n}{n - o(n)} \sim 2^n, \\ L^{(T)}(n, q_0) &\lesssim 2^k + 2^{n-k} + n2^{n-k} \sim 2^k + n2^{n-k}, \\ L^{(T)}(n, q_0) &\lesssim 2^{\lceil n/\phi(n) \rceil} + \frac{n2^n}{2^{\lceil n/\phi(n) \rceil}} \sim \frac{n2^n}{2^{\lceil n/\phi(n) \rceil}}. \end{aligned}$$

From these upper bounds and the equation ((3.1)) the upper bound for the function $W(n, q_0)$ from the theorem follows. \square

Note that in the case, when $W^{(T)} = O(W^{(C)}) = \text{const}$, the following equation holds:

$$W(n, q_0) \asymp L^{(C)}(n, q_0) \sim L(n, q_0) ,$$

where $q_0 \sim n2^{n - \lceil n / \phi(n) \rceil}$; $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions. In other words, the number of C²NOT gates in a reversible circuit produced by the synthesis algorithm **A1** is negligible compared to the overall gate complexity. And from the equations ((3.2)), ((3.4)), ((3.5)) and ((3.7)) it follows that in a reversible circuit without additional inputs the number of C²NOT gates is equivalent by the order of magnitude to the overall gate complexity.

5. Reducing the depth with the help of additional inputs

We described the synthesis algorithm **A1**, whose main goal was the gate complexity reduction with the help of additional inputs. However, we can use a similar technique to reduce the depth of a reversible circuit. Let's denote such an algorithm as **A2**. The essence of the synthesis algorithm **A2** is the copying of the value from an output to the additional inputs with the logarithmic depth (see Figure 5.1). After this we can perform a desired operation with the depth equal to one. All we need to do is to copy the value a sufficient number of times.

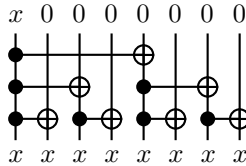


Figure 5.1: Copying the value x to the additional inputs with the logarithmic depth.

First, we prove the following lemma about the depth of a reversible circuit implementing all conjunctions of n variables of the form $x_1^{a_1} \wedge \dots \wedge x_n^{a_n}$, $a_i \in \mathbb{B}$.

LEMMA 5.1. *All conjunctions of n variables of the form $x_1^{a_1} \wedge \dots \wedge x_n^{a_n}$, $a_i \in \mathbb{B}$, can be implemented in a reversible circuit \mathfrak{S}_n , which consists of gates from Ω_{n+q}^2 , with the depth $D(\mathfrak{S}_n) \sim n$, the gate complexity $L(\mathfrak{S}_n) \sim 3 \cdot 2^n$ and with $q(\mathfrak{S}_n) \sim 3 \cdot 2^n$ additional inputs.*

PROOF. First step is obtaining inversions of all input variables: \bar{x}_i , $1 \leq i \leq n$. This can be done with the depth $D_1 = 2$, using $L_1 = 2n$ NOT and CNOT gates and $q_1 = n$ additional inputs.

We construct our reversible circuit \mathfrak{S}_n in the same way as in Lemma 4.1, using sub-circuits $\mathfrak{S}_{\lceil n/2 \rceil}$ and $\mathfrak{S}_{\lfloor n/2 \rfloor}$ (see Figure 5.2). Any output of these sub-circuits will be used no more than in $2 \cdot 2^{n/2}$ conjunctions, so all conjunctions can be implemented with the depth $D_2 \leq 2 + n/2$, using 2^{n+1} CNOT gates, 2^n C²NOT gates and $q_2 = 3 \cdot 2^n$ additional inputs.

Hence, the following equations hold:

$$\begin{aligned} D(\mathfrak{S}_n) &\sim \frac{n}{2} + D(\mathfrak{S}_{n/2}) \sim n, \\ L(\mathfrak{S}_n) &\sim 3 \cdot 2^n + 2L(\mathfrak{S}_{n/2}) \sim 3 \cdot 2^n, \\ q(\mathfrak{S}_n) &\sim 3 \cdot 2^n + 2q(\mathfrak{S}_{n/2}) \sim 3 \cdot 2^n. \end{aligned}$$

□

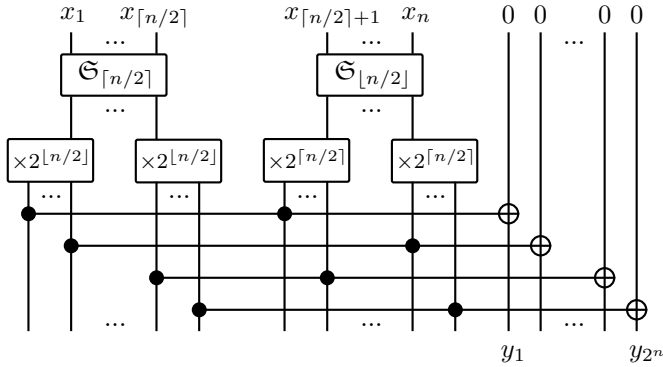


Figure 5.2: The structure of a reversible circuit \mathfrak{S}_n implementing all conjunctions of n variables with the minimal depth.

Now we can prove the next theorem of the paper.

THEOREM 5.2.

$$D(n, q_1) \lesssim 3n, \text{ if } q_1 \sim 2^n.$$

A reversible circuit \mathfrak{S} with the depth $D(\mathfrak{S}) \sim 3n$ has the gate complexity $L(\mathfrak{S}) \sim 2^{n+1}$ and the quantum weight $W(\mathfrak{S}) \sim W^{(C)} \cdot 2^{n+1} + W^{(T)} \cdot n2^{n-\lceil n/\phi(n) \rceil}$, where $\phi(n) \leq n/(\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions.

PROOF. We will describe the synthesis algorithm **A2**, which is similar to the synthesis algorithm **A1** and whose main goal is the reduction of the depth with the help of additional inputs.

Let's consider a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^n$. It can be represented by the formulae ((4.3))–((4.5)), see pages 11–12.

Note that all Boolean functions of a group G_i can be implemented, using a similar technique as in the Lemma 5.1. This requires $L \sim 3 \cdot 2^s$ CNOT gates (2^{s+1} gates are used to copy values to the additional inputs) and $q \sim 2^{s+1}$ additional inputs (see Figure 5.3).

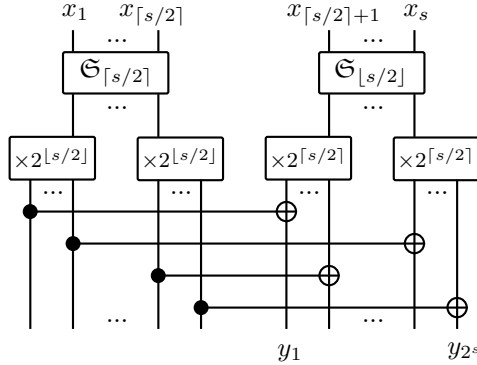


Figure 5.3: The structure of a reversible circuit implementing all Boolean functions $g \in G_i$ with the minimal depth.

The synthesis algorithm **A2** constructs a reversible circuit \mathfrak{S} implementing the transformation f ((4.3)) from the following sub-circuits (see Figure 5.4):

1. Sub-circuit \mathfrak{S}_1 implementing all conjunctions of the first k variables x_i by the Lemma 5.1 with the depth $D_1 \sim k$, the

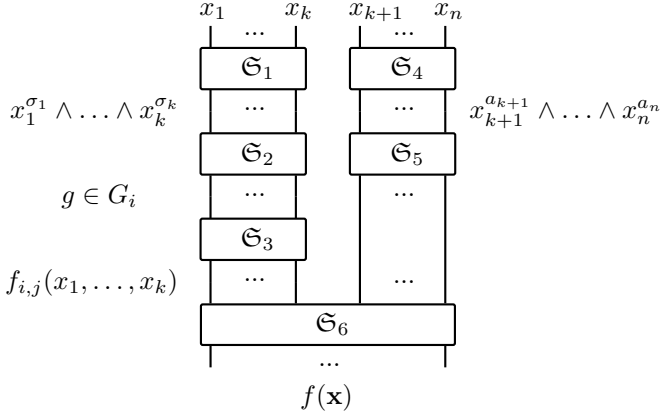


Figure 5.4: The structure of a reversible circuit \mathfrak{S} produced by the synthesis algorithm **A2**.

- gate complexity $L_1 \sim 3 \cdot 2^k$ and with $q_1 \sim 3 \cdot 2^k$ additional inputs. The sub-circuit contains 2^k C²NOT gates.
2. Sub-circuit \mathfrak{S}_2 implementing all Boolean functions $g \in G_i$ for all $i \in \mathbb{Z}_p$ by the formula ((4.4)) with the depth $D_2 \sim s$, the gate complexity $L_2 \sim 3p2^s$ and with $q_2 \sim p2^{s+1}$ additional inputs (see the note above about the implementation of all Boolean functions of a group G_i). The sub-circuit consists only of CNOT gates.
 3. Sub-circuit \mathfrak{S}_3 implementing all $n2^{n-k}$ coordinate functions $f_{i,j}(\mathbf{x})$, $i \in \mathbb{Z}_{2^{n-k}}$, $j \in \mathbb{Z}_n$, by the formula ((4.5)). The feature of this sub-circuit is that a Boolean function $g \in G_t$ can be used more than once. The maximum usage count for a function g is $n2^{n-k}$. So, first of all we need to copy the values from outputs of the sub-circuit \mathfrak{S}_2 for all such Boolean functions. This can be done with the depth equal to $\lceil n - k + \log_2 n \rceil$, using no more than $pn2^{n-k}$ gates and $pn2^{n-k}$ additional inputs (see Figure 5.1). After this, we implement XOR function of obtained outputs with the depth equal to $\lceil \log_2 p \rceil$, the gate complexity equal to $(p - 1)n2^{n-k}$ and without additional inputs (see Figure 5.5). Therefore, the sub-circuit \mathfrak{S}_3 has the depth $D_3 \sim n - k + \log_2 p$, the

gate complexity $L_3 \sim (2p - 1)n2^{n-k}$ and $q_3 \sim (p - 1)n2^{n-k}$ additional inputs. It consists only of CNOT gates.

4. Sub-circuit \mathfrak{S}_4 implementing all conjunctions of the last $(n - k)$ variables x_i by the Lemma 5.1 with the depth $D_4 \sim (n - k)$, the gate complexity $L_4 \sim 3 \cdot 2^{n-k}$ and with $q_4 \sim 3 \cdot 2^{n-k}$ additional inputs. The sub-circuit contains 2^{n-k} C²NOT gates.
5. Sub-circuit \mathfrak{S}_5 , which is needed to copy $(n - 1)$ times every output of the sub-circuit \mathfrak{S}_4 . This can be done with the depth $D_5 \sim \log_2 n$, the gate complexity $L_5 = (n - 1)2^{n-k}$ and with $q_5 = (n - 1)2^{n-k}$ additional inputs. The sub-circuit consists only of CNOT gates.
6. Sub-circuit \mathfrak{S}_6 implementing the transformation f by the formula ((4.3)). The structure of the sub-circuit is as follows: all $n2^{n-k}$ coordinate functions $f_{i,j}(\mathbf{x})$ are grouped by 2^{n-k} functions (n groups in total, which correspond to n outputs of the transformation f). Functions in a group are again grouped by two. For every pair of functions we implement a conjunction of the corresponding outputs of sub-circuits \mathfrak{S}_3 and \mathfrak{S}_5 , using 2 C²NOT gates and one additional input for storing an intermediate result (see Figure 5.6). Thus, this part of the sub-circuit has the depth equal to 2, requires $n2^{n-k}$ C²NOT gates and $n2^{n-k-1}$ additional inputs. After this, in every of n groups of obtained outputs we implement XOR function with the logarithmic depth (see Figure 5.5 and Figure 5.6). This part of the sub-circuit has the depth equal to $n - k - 1$, requires $n(2^{n-k-1} - 1)$ CNOT gates and doesn't require additional inputs, because we can use the existing outputs.

Summing up, the sub-circuit \mathfrak{S}_6 has the depth $D_6 \sim n - k$, the gate complexity $L_6 \sim 3n2^{n-k-1}$ and $q_6 \sim n2^{n-k-1}$ additional inputs.

Note that the sub-circuits \mathfrak{S}_1 – \mathfrak{S}_3 and \mathfrak{S}_4 – \mathfrak{S}_5 can work in parallel, because they use disjoint subsets of the inputs x_1, \dots, x_n .

We are seeking the values of parameters k and s that satisfy

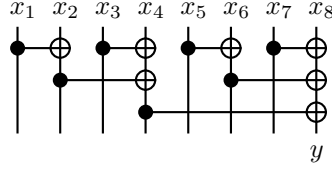


Figure 5.5: Implementing the function $y = x_1 \oplus \cdots \oplus x_8$ in a reversible circuit with the logarithmic depth (this is a part of the sub-circuit \mathfrak{S}_3 produced by the synthesis algorithm **A2**).

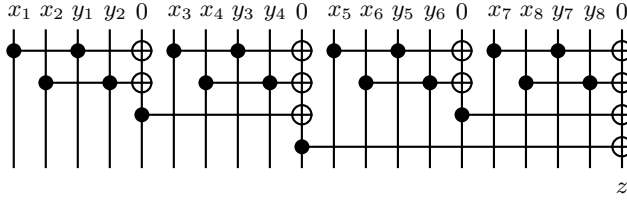


Figure 5.6: Implementing the function $z = \bigoplus_{i=1}^8 x_i \wedge y_i$ in a reversible circuit with the logarithmic depth (this is a part of the sub-circuit \mathfrak{S}_6 produced by the synthesis algorithm **A2**).

the following conditions:

$$\begin{cases} k + s = n, \\ 1 \leq k < n, \\ 1 \leq s < n, \\ 2^k / s \geq \psi(n), \text{ where } \psi(n) \text{ is a growing function.} \end{cases}$$

In this case $p = \lceil 2^k / s \rceil \sim 2^k / s$.

Summing up depths, gate complexities and the number of additional inputs for all sub-circuits \mathfrak{S}_1 – \mathfrak{S}_6 , we obtain the following bounds for the circuit \mathfrak{S} parameters.

The depth:

$$\begin{aligned} D(\mathfrak{S}) &\sim \max(k + s + n - k + \log_2 p; n - k + \log_n) + n - k, \\ (5.3) \quad D(\mathfrak{S}) &\sim 2n + s. \end{aligned}$$

The gate complexity:

$$\begin{aligned} L(\mathfrak{S}) &\sim 3 \cdot 2^k + 3p2^s + (2p - 1)n2^{n-k} + \\ &\quad + 3 \cdot 2^{n-k} + n2^{n-k} + 3n2^{n-k-1}, \end{aligned}$$

$$(5.4) \quad L(\mathfrak{S}) \sim 3 \cdot \frac{2^n}{2^s} + \frac{3 \cdot 2^n}{s} + \frac{n2^{n+1}}{s} \sim \frac{n2^{n+1}}{s}.$$

The number of additional inputs:

$$(5.5) \quad \begin{aligned} q(\mathfrak{S}) &\sim 3 \cdot 2^k + p2^{s+1} + pn2^{n-k} + 3 \cdot 2^{n-k} + n2^{n-k} + n2^{n-k-1}, \\ q(\mathfrak{S}) &\sim 3 \cdot \frac{2^n}{2^s} + \frac{2^{n+1}}{s} + \frac{n2^n}{s} \sim \frac{n2^n}{s}. \end{aligned}$$

From the description of the synthesis algorithm **A2** we can see that

$$\begin{aligned} L_1^{(C)} &\sim 2^{k+1}, & L_1^{(T)} &\sim 2^k, \\ L_2^{(C)} &\sim 3p2^s, & L_2^{(T)} &= 0, \\ L_3^{(C)} &\sim pn2^{n-k+1}, & L_3^{(T)} &= 0, \\ L_4^{(C)} &\sim 2^{n-k+1}, & L_4^{(T)} &\sim 2^{n-k}, \\ L_5^{(C)} &\sim n2^{n-k}, & L_5^{(T)} &= 0, \\ L_6^{(C)} &\sim n2^{n-k-1}, & L_6^{(T)} &= n2^{n-k}. \end{aligned}$$

This implies that

$$(5.6) \quad L^{(C)}(\mathfrak{S}) \sim 2^{k+1} + \frac{n2^{n+1}}{s} \sim \frac{n2^{n+1}}{s},$$

$$(5.7) \quad L^{(T)}(\mathfrak{S}) \sim 2^k + n2^{n-k}.$$

Let $k = \lceil n / \phi(n) \rceil$, where $\phi(n) < n$ is a growing function. In this case $s = n - \lceil n / \phi(n) \rceil$ and

$$\begin{aligned} 2^k \geq s\psi(n) &\Rightarrow k \geq \log_2 s + \log_2 \psi(n) \Rightarrow \\ &\Rightarrow \phi(n) \leq \frac{n}{\log_2 s + \log_2 \psi(n) - 1}. \end{aligned}$$

We can choose any arbitrarily slowly growing functions $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$.

Hence, we obtain the following bounds for the circuit \mathfrak{S} param-

eters:

$$\begin{aligned}
 D(\mathfrak{S}) &\sim 2n + n - \lceil n / \phi(n) \rceil \sim 3n , \\
 L(\mathfrak{S}) &\sim L^{(C)}(\mathfrak{S}) \sim \frac{n2^{n+1}}{n - \lceil n / \phi(n) \rceil} \sim 2^{n+1} , \\
 L^{(T)}(\mathfrak{S}) &\sim 2^{\lceil n / \phi(n) \rceil} + n2^{n - \lceil n / \phi(n) \rceil} \sim n2^{n - \lceil n / \phi(n) \rceil} , \\
 q(\mathfrak{S}) &\sim \frac{n2^n}{n - \lceil n / \phi(n) \rceil} \sim 2^n .
 \end{aligned}$$

Since the synthesis algorithm **A2** can produce a reversible circuit \mathfrak{S} for any Boolean transformation $f \in F(n, q)$, we can state that $D(n, q_1) \leq D(\mathfrak{S}) \sim 3n$, if $q_1 \sim 2^n$.

Also we can state that $L(\mathfrak{S}) \sim 2^{n+1}$ and $W(\mathfrak{S}) \sim W^{(C)} \cdot 2^{n+1} + W^{(T)} \cdot n2^{n - \lceil n / \phi(n) \rceil}$, where $\phi(n) \leq n / (\log_2 n + \log_2 \psi(n))$ and $\psi(n)$ are arbitrarily slowly growing functions. \square

Finally, we prove the last theorem of the paper.

THEOREM 5.8.

$$D(n, q_2) \lesssim 2n, \text{ if } q_2 \sim \phi(n)2^n ,$$

where $\phi(n) < n$ is an arbitrarily slowly growing function. A reversible circuit \mathfrak{S} with the depth $D(\mathfrak{S}) \sim 2n$ has the gate complexity $L(\mathfrak{S}) \sim \phi(n)2^{n+1}$ and the quantum weight $W(\mathfrak{S}) \sim W^{(C)} \cdot \phi(n)2^{n+1} + W^{(T)} \cdot 2^{n - \lceil n / \phi(n) \rceil}$.

PROOF. Proof is based on the proof of the previous theorem.

Let $s = \lceil n / \phi(n) \rceil$, where $\phi(n) < n$ is a growing function. In this case $k = n - \lceil n / \phi(n) \rceil$ and

$$\psi(n) \leq \frac{2^k}{s} \leq \frac{\phi(n)2^{n - o(n)}}{n} .$$

We can see that we always able to choose a growing function $\psi(n)$ for any growing function $\phi(n) < n$.

From the equations ((5.3))–((5.7)) it follows that for these values of k and s the following equations hold:

$$\begin{aligned} D(\mathfrak{S}) &\sim 2n + \lceil n / \phi(n) \rceil \sim 2n , \\ L(\mathfrak{S}) &\sim L^{(C)}(\mathfrak{S}) \sim \frac{n2^{n+1}}{\lceil n / \phi(n) \rceil} \sim \phi(n)2^{n+1} , \\ L^{(T)}(\mathfrak{S}) &\sim 2^{n-\lceil n / \phi(n) \rceil} + n2^{\lceil n / \phi(n) \rceil} \sim 2^{n-\lceil n / \phi(n) \rceil} , \\ q(\mathfrak{S}) &\sim \frac{n2^n}{\lceil n / \phi(n) \rceil} \sim \phi(n)2^n . \end{aligned}$$

Since the synthesis algorithm **A2** can produce a reversible circuit \mathfrak{S} for any Boolean transformation $f \in F(n, q)$, we can state that $D(n, q_2) \leq D(\mathfrak{S}) \sim 2n$, if $q_2 \sim \phi(n)2^n$, where $\phi(n) < n$ is an arbitrarily slowly growing function.

Also, $L(\mathfrak{S}) \sim \phi(n)2^{n+1}$ and $W(\mathfrak{S}) \sim W^{(C)} \cdot \phi(n)2^{n+1} + W^{(T)}$. \square

6. Conclusion

We have discussed the problem of the general synthesis of a reversible circuit, which consists of NOT, CNOT and C²NOT gates and which has additional inputs, with the lowest possible gate complexity and depth. We have studied the Shannon gate complexity function $L(n, q)$, the depth function $D(n, q)$ and the quantum weight function $W(n, q)$ for a reversible circuit with additional inputs, which implements a transformation $f: \mathbb{B}^n \rightarrow \mathbb{B}^n$ from the set $F(n, q)$.

The main result of the paper is the following claim.

CLAIM 6.1. *The usage of additional memory in a reversible circuit consisting of NOT, CNOT and C²NOT gates almost always allows to reduce its gate complexity, the depth and the quantum weight.*

The proof of the claim follows from the Theorems Theorem 4.2–Theorem 5.8 and the lower bounds ((3.2))–((3.4)).

Solving the problem of the reversible logic synthesis, one should find a compromise between the gate complexity, the depth (working

time) and the amount of used memory (additional inputs) of a reversible circuit. Unfortunately, we were not able to establish good upper and lower bounds for the depth function $D(n, q)$, which would be asymptotically equivalent to each other by the order of magnitude. However, the obtained bounds are sufficient to prove the main claim of the paper.

Further research should establish more precise relationship of a reversible circuit's parameters from the number of additional inputs in the circuit. We hope that the paper will be the first step in this direction.

Acknowledgements

The reported study was partially supported by RFBR, research project No. 16-01-00196 A.

References

- NABILA ABDESSAIED, ROBERT WILLE, MATHIAS SOEKEN & ROLF DRECHSLER (2013). Reducing the Depth of Quantum Circuits Using Additional Circuit Lines. In *Reversible Computation*, GERHARD W. DUECK & D. MICHAEL MILLER, editors, volume 7948 of *Lecture Notes in Computer Science*, 221–233. Springer Berlin Heidelberg. ISBN 978-3-642-38985-6.
- ADRIANO BARENCO, CHARLES H. BENNETT, RICHARD CLEVE, DAVID P. DIVINCENZO, NORMAN MARGOLUS, PETER SHOR, TYCHO SLEATOR, JOHN A. SMOLIN & HARALD WEINFURTER (1995). Elementary gates for quantum computation. *Phys. Rev. A* **52**(5), 3457–3467.
- C. H. BENNETT (1973). Logical Reversibility of Computation. *IBM Journal of Research and Development* **17**(6), 525–532. ISSN 0018-8646.
- RICHARD P. FEYNMAN (1985). Quantum Mechanical Computers. *Optics News* **11**(2), 11–20.
- N. A. KARPOVA (1987). On Complexity of Computations with Limited Memory. In *FCT*, LOTHAR BUDACH, RAIS GATIC BAKHARAJEV & OLEG BORISOVIC LIPANOV, editors, volume 278 of *Lecture Notes in Computer Science*, 234–235. Springer. ISBN 3-540-18740-5.

ANDREI B. KHLOPOTINE, MAREK A. PERKOWSKI & PAWEŁ KERN-TOPI (2002). Reversible Logic Synthesis by Iterative Compositions. In *IWLS*, 261–266.

V. M. KHRAPCHENKO (1995). New inequality relations between depth and delay. *Diskr. Mat.* **7**(4), 77–85. In Russian.

R. LANDAUER (1961). Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development* **5**(3), 183–191. ISSN 0018-8646.

O. B. LUPANOV (1970). On Circuits of Functional Elements with Delays. In *Probl. Kibernet.*, volume 23, 43–81. Nauka Publishers, Moscow. In Russian.

D. A. MASLOV, G. W. DUECK & D. M. MILLER (2007). Techniques for the Synthesis of Reversible Toffoli Networks. *ACM Trans. Des. Autom. Electron. Syst.* **12**(4). ISSN 1084-4309.

D. M. MILLER & G. W. DUECK (2003). Spectral Techniques for Reversible Logic Synthesis. In *6th International Symposium on Representations and Methodology of Future Computing Technologies*, 56–62.

D. M. MILLER, R. WILLE & R. DRECHSLER (2010). Reducing Reversible Circuit Cost by Adding Lines. In *Multiple-Valued Logic (ISMVL), 2010 40th IEEE International Symposium on*, 217–222. ISSN 0195-623X.

D. MICHAEL MILLER, DMITRI A MASLOV & GERHARD W. DUECK (2003). A Transformation Based Algorithm for Reversible Logic Synthesis. In *Proceedings of the 40th Annual Design Automation Conference, DAC '03*, 318–323. ACM, New York, NY, USA. ISBN 1-58113-688-9.

M. SAEEDI, M. SEDIGHI & M. S. ZAMANI (2007). A novel synthesis algorithm for reversible circuits. In *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, 65–68. ISSN 1092-3152.

MEHDI SAEEDI, MORTEZA SAHEB ZAMANI, MEHDI SEDIGHI & ZAHRA SASANIAN (2010). Reversible Circuit Synthesis Using a Cycle-based Approach. *J. Emerg. Technol. Comput. Syst.* **6**(4), 13:1–13:26. ISSN 1550-4832.

CLAUDE E. SHANNON (1949). The Synthesis of Two Terminal Switching Circuits. *Bell System Technical Journal* **28**(1), 59–98. ISSN 0005-8580.

V. V. SHENDE, A. K. PRASAD, I. L. MARKOV & J. P. HAYES (2003). Synthesis of reversible logic circuits. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* **22**(6), 710–722. ISSN 0278-0070.

TOMMASO TOFFOLI (1980). Reversible computing. In *Automata, Languages and Programming*, JACO DE BAKKER & JAN VAN LEEUWEN, editors, volume 85 of *Lecture Notes in Computer Science*, 632–644. Springer Berlin Heidelberg.

DMITRY V. ZAKABLUKOV (2014). Fast Synthesis of Invertible Circuits Based on Permutation Group Theory. *Prikl. Diskr. Mat.* **24**(2), 101–109. In Russian.

DMITRY V. ZAKABLUKOV (2015). On Asymptotic Gate Complexity and Depth of Reversible Circuits Without Additional Memory. *ArXiv e-prints* URL <http://arxiv.org/abs/1504.06876>.

Manuscript received 19 March 2016

DMITRY V. ZAKABLUKOV
Department of Information Security
Bauman Moscow State Technical
University
Moscow, Russian Federation
105005
dmitriy.zakablukov@gmail.com